Multidimensional Context-Aware Adaptation of Service Front-Ends

Project no. FP7 – ICT – 258030

Deliverable 2.1.2
CARF and CADS (R2)

Due date of deliverable: 29/02/2012
Actual submission to EC date: 29/02/2012

Project co-funded by the European Commission within the Seventh Framework Programme (2007-2013)

| Dissemination level | [PU] | [Public] | Yes |

This work is licensed under a Creative Commons Attribution-Noncommercial-Share Alike 3.0 License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-sa/3.0/ or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA (This license is only applied when the deliverable is public).
Executive Summary

The main goal of this deliverable is to present the evolution process of the two theoretical methods that together with the CARFO Ontology compose the framework for context-aware multidimensional adaptation: the Context-aware Design Space (CADS) and the Context-aware Reference Framework (CARF).

These methods are complementary and represent a unified approach to present concepts that are related to context-aware adaptation. They consist in graphical representations that organize relevant dimensions for adaptation and adaptation techniques.

This deliverable presents the discussion and evolution of these theoretical methods based on their preliminary versions. Besides we present a critical analysis of the methods, propose solutions for their trade-offs and we conclude it by describing examples of use and application of these methods.
## Table of Contents

1 Introduction .................................................................................................................................................. 6
  1.1 Objectives ........................................................................................................................................ 6
  1.2 Audience ........................................................................................................................................... 6
  1.3 Related documents .......................................................................................................................... 6
  1.4 Organization of this document ....................................................................................................... 6

2 Description of Work .................................................................................................................................. 7
  2.1 Motivation .......................................................................................................................................... 7
  2.2 Goal .................................................................................................................................................... 7
  2.3 Description ....................................................................................................................................... 7

3 Context-Aware Design Space (CADS) ..................................................................................................... 9
  3.1 Contextualization .............................................................................................................................. 9
  3.2 Description ....................................................................................................................................... 12
  3.3 Application ...................................................................................................................................... 13
  3.4 Advantages ..................................................................................................................................... 14
  3.5 Final Remarks ................................................................................................................................ 15

4 Context-Aware Reference Framework (CARF) .................................................................................... 16
  4.1 Contextualization ............................................................................................................................. 16
  4.2 Description ....................................................................................................................................... 17
  4.3 How .................................................................................................................................................. 18
    4.3.1 Adaptation Techniques and Methods ......................................................................................... 18
    4.3.2 Adaptation Strategies ............................................................................................................... 18
  4.4 To What ......................................................................................................................................... 21
    4.4.1 User .......................................................................................................................................... 21
    4.4.2 Platform .................................................................................................................................... 21
    4.4.3 Environment ............................................................................................................................. 22
  4.5 Where ............................................................................................................................................. 22
  4.6 Why ................................................................................................................................................ 23
  4.7 What .............................................................................................................................................. 23
  4.8 Who ............................................................................................................................................... 23
  4.9 When ............................................................................................................................................. 23
  4.10 Application .................................................................................................................................... 24
    4.10.1 Applied CARF ......................................................................................................................... 24
  4.11 Final Remarks ............................................................................................................................... 24

5 Conclusion .............................................................................................................................................. 25
  5.1 Final Remarks .................................................................................................................................. 25
  5.2 Future Work ..................................................................................................................................... 25

References .................................................................................................................................................... 26

Acknowledgements .................................................................................................................................... 27
1 Introduction

1.1 Objectives
This deliverable describes a more consolidated version of the CADS and the CARF: the two main components for the development of the framework for adaptation of services front-end according to context awareness. CADS stands for Context-Aware Design Space, a graphical representation that provides an overview of the possible dimensions for adaptation and their respective granularity levels. CARF consists of a Context-Aware Reference Framework that refines and details the dimensions of the CADS. In a first moment, an extensive investigation was performed in order to identify and describe in details the adaptation elements and the related techniques, then the methods were analysed, their trade-offs identified, and solutions proposed. These methodologies are complementary and provide the main components for the framework aimed by the Serenoa Project.

1.2 Audience
The target audience consists of researchers, industry.

1.3 Related documents
- D1.1.1 and D1.1.2 – Requirements: The theoretical methods are connected to the list of Requirements of Serenoa, specially concerning the non functional requirements and the branch why of the CARF that considers software qualities
- D2.1.1 – CADS and CARF: The preliminary versions and the definitions of the CADS and the CARF can be found in the first release of the deliverable
- D2.2.1 and D2.3.1– CARFO: The CARFO, together with the CADS and the CARF, compose the computational framework for Serenoa. Besides, the CARFO implementation is based on concepts defined for the CADS and the CARF
- D2.4.1 – The Criteria for Evaluation of CAA also can be found in the branch why of the CARF
- D3.1.1 – The Reference Models are based on concepts defined for the CARF and the CADS (specially concerning the meta model for context-aware adaptation)
- D4.2.1 – The Library of Algorithms of Serenoa was built based of adaptation techniques described as templates and defined by the how branch of the CARF

1.4 Organization of this document
Chapter 1 presents the goal, audience and related documentation with this deliverable. Chapter 2 describes the work done, as well as their motivations and goals. Chapter 3 describes and exemplify de Design Space, highlighting the evolution of the model. In Chapter 4, the Reference Framework is described and illustrated with examples. Chapter 5 presents the concluding remarks for this deliverable.
2 Description of Work

2.1 Motivation

An application can be adapted according to many different aspects, including for instance information regarding its user, platform and environment. These are the main context information, but not only they have to be considered, and also the dimensions in which adaptation may be applied.

Nowadays stakeholders who wish to develop context-aware adaptation do not have available theoretical methodologies and online tools that support their work during the complete development life-cycle of applications. And thus they must normally rely on existing publications, which do not provide a broad overview about the implementation possibilities. Furthermore, there is no publication concerning methods to evaluate the coverage level of adaptive and adaptable applications.

Aiming to organize concepts related to context-aware adaptation, we proposed and created two theoretical methods. These methods permit stakeholders not only to identify possibilities for implementation but also to analyse and to evaluate applications that perform context-aware adaptation. In the sense, the Context-aware Design Space (CADS) and the Context-aware Reference Framework (CARF) were created as two complementary representations that, together with the CARFO Ontology, compose the main framework for the Serenoa Project.

The first release of D2.1.1 concerned the proposition and creation of these methods; this second release though tackles their evaluation, discussion, analysis and evolution.

2.2 Goal

In order to devise a computational framework for (multidimensional) context-aware adaptation of SFEs three methods have been adopted:

- a Reference Framework (CARF) identifying the relevant abstraction levels for the description of SFEs sensitive to the multiple dimensions of the context of use.
- a Design Space (CADS) identifying the relevant design options and how they can vary to accommodate different scenarios and requirements.
- an Ontology (CARFO) for Multi-Dimensional Adaptation of SFEs which expresses formal definitions for the more relevant concepts in both CADS and CARF, and the relationships between them. CARFO will work as a common vocabulary for the different module and will allow some reasoning on knowledge bases built following the knowledge model of such ontology.

The CARF and the CADS provide a theoretic and conceptual perspective as a unifying approach. We consider adaptations that take into account different contextual aspects (and their mutual influences) at the same time (multi-dimensional adaptation), thus leveraging existing work on adaptive and multi-target UIs.

2.3 Description

Task 2.1 Reference Framework (CARF) and Design Space (CADS) for CAA of SFEs [led by UCL]

This task is devoted to devise a sound definition of two main components of the Computational Framework and their relationships:

1. A Reference Framework for CAA of SFEs. The starting point to produce the CARF is the Cameleon Reference Framework [Calvary et al., 2002], which is being leveraged to take into account new technological developments such as Rich Internet Applications (Web 2.0) and Distributed User Interfaces. For this purpose, a conceptual map defining dimensions for context-aware adaptation of SFEs will be progressively built along the following axes: how, when, what, with respect to what, where, etc. This CARF will be fed by already existing surveys on adaptation and will progressively evolve by considering more advanced adaptation techniques. The conceptual
map will also be implemented as visualisation software based on hyperbolic trees in order to check consistency, to reason about it, etc.

2. A Design Space for multi-dimensional CAA of SFEs: The CADS will also introduce the basic principles that drive context-aware SFEs, enabling to explore different adaptation methods that will be developed during the project. The starting point to produce the CADS will be the FP 6 Similar Adaptation Space that will be expanded and refined.
3 Context-Aware Design Space (CADS)

The context-aware design space is a theoretical method that provides stakeholders a tool to support them in the phases of implementation, analysis and evaluation of adaptive and adaptable applications. The goal of CADS is helping developers before implementing their applications to be aware of possible dimensions and granularity levels for performing adaptation, and after the implementation to analyse, evaluate and compare these dimensions regarding their respective coverage levels. As such the CADS supports the analysis and the comparison of different applications that execute adaptation and during their complete development life-cycle.

The CADS is built as a radar chart\(^1\), which is a useful approach to represent multi variable observations with an arbitrary number of variables. In principle, these representations are used for ordinal measurements, however in the CADS case qualitative values are represented with their respective empiric scale associated.

3.1 Contextualization

The previous version of the CADS was composed by 7 dimensions, namely: adaptation means, UI component granularity, state recovery granularity, UI deployment, context of use, technological space coverage and existence of a meta-UI. The detailed definition of these concepts is available in the D2.1.1.

The graphic representation of these dimensions in orthogonal axis, with colours, allows the analysis and evaluation of adaptive applications in a unified view, i.e. all dimensions start from a central point and are comprised in the same diagram. Developers appreciate this fact, as we could identify in a short survey applied later. By drawing the curves in the corresponding points for each dimension, the CADS allowed the comparison of two or more applications, in the same view, permitting to detect for instance dimensions that were underexplored and that could be better covered in other aspects too.

In [Desruelle et al., 2011] the previous version of CADS was applied to identify the required degree of application adaptability aimed by the webinos\(^2\) platform (as Figure 1 illustrates).

![Figure 1. Application of the CADS [Desruelle et al., 2011]](image)

The CADS has many benefits, among which we can mainly highlight: it is exploratory, descriptive, extensible and flexible. However, by evaluating its application we could clearly identify some issues.

After presenting and discussing the previous version of CADS with Serenoa members and reviewers, we concluded that representing the dimension axes with arrowed lines denoted a sequential relation between the internal granularity levels of each dimension, however such interpretation could not be applied for all the

---

\(^1\) http://en.wikipedia.org/wiki/Radar_chart

\(^2\) Webinos is a EU-funded project aiming to define and deliver an Open Source Platform and software components for the future internet (http://webinos.org/)
axes previously represented, which could consequently lead to misunderstandings. By taking into account for instance the dimension of the context of use, one is not able to assure that considering the platform characteristics represents a better (or worse) approach than considering the environmental ones. Besides, one could not infer that by considering the environmental characteristics, platforms and users characteristics have equally been considered. In an initial attempt to solve this issue, the first version of the CADS was updated. In this newer version, illustrated by Figure 2, the arrows were replaced by circles for unordered dimensions. This solution tackled the misinterpretation problem of each dimension, however it could not solve the issue of applying the CADS for analysing and comparing multiple adaptive systems at the same time (since the unordered dimensions were still comprised in the CADS). Besides this, the applied version of the CADS could have a negative impact regarding the readability of the dimensions’ labels, as the Figure 2 (on the right diagram) illustrates.

![Figure 2. First version of the CADS, updated without arrows, and applied CADS illustrating an application](image)

Given the issues mentioned above, the weaknesses of the method were detected, discussed, and a new proposal for a second version of the CADS was created. In this second version, illustrated by Figure 3, the vertical axes represented just the ordered dimensions and the horizontal axes represented only the unordered dimensions. By drawing curves between the vertical axes, stakeholders were able to analyse and compare multiple applications; in this manner the coverage level for each dimension could be easily identified. One possible approach to analyse the horizontal dimensions, i.e. unordered ones, consisted in marking, with an $X$, each concept that was previously considered for the application under analysis. To perform additional analysis, multiple marks could be added in each dimension level. Another alternative approach consisted in replacing the horizontal axis by rectangles and filling them with background colours (or texture) until reaching the levels that were considered by the applications of interest.

Aiming to verify the developers’ opinion and their acceptance of this version, a short survey was created and applied with 5 subjects. The subjects have experience with research and development, background in HCI domain and adaptation, and they belong to the scientific field. This survey was defined and published using SurveyMonkey. This survey contained the first and the second version of CADS. As a result, surprisingly, the subjects who replied the survey were unanimous in assessing the unified representation with orthogonal axis as the best one (radar chart). They believe it is clearer and more intuitive to look at all dimensions simultaneously if they are all grouped.

Therefore, knowing that only concepts that can be ordered should be represented by an arrowed axis, and that the unified view was considered more intuitive in terms of semantic interpretation for developers, a third version of the CADS was created, comprising only adaptation dimensions that have ordered levels (in terms of granularity for instance). This third version maintained the unified representation, and tackled the issue with unordered dimensions, by transferring them to the CARF diagram (described in Section 4).

---

3 Survey Monkey is an online tool available at: [www.surveymonkey.com/](http://www.surveymonkey.com/) for creating, publishing and applying surveys.
Figure 3. Second version of the CADS, dimensions are apart according to their order levels.

Four colours were chosen to reinforce the order relationships of the CADS dimensions, however, since this information is also represented by the spatial representation of concentric circles with varying diameters, accessibility issues were prevented.

This new version of the CADS conserved the main advantages of the previous diagram. Therefore, the virtues proposed by Lafon (2000) for design spaces, are still valid, as such, the CADS is considered comparative since multiple applications can be analysed according to the same criteria; exploratory, since each dimensions can be analysed in terms of exploration, i.e. identifying additional opportunities for extensions; and descriptive, since the dimensions are precisely pre-defined, in a consistent and unique way. Furthermore, the CADS represents also an approach that is extensible, since new dimensions can be added or better refined and flexible, since dimensions can be removed or added permitting a more focused analysis.

Clearly, the interpretation of scales for the dimensions chosen can vary according to the context. However, it is a general interpretation that is provided by the CADS. Once the concepts cannot (in principle) be numerically evaluated and compared, it is their semantic meanings and interpretations that must be taken into account. The proportions are also empirically associated with the dimensions, since no formal experiments were performed so far to identify real metrics that could be then associated with each dimension and its respective granularity levels. For each case of application of the CADS, its use must be defined and discussed. It is worth to notice that all CADS dimensions, although comprised in the same representation, are still independent, and as such the concentric circles while aid the comparison of different granularity levels, do not necessarily represent the same coverage level between two different dimensions.

The central circle of the CADS, coloured in red, represents the absence of adaptation concerns, for example when no adaptation process is performed an application can be classified as designed regarding its autonomy level. For each subsequent circle an additional level of adaptation coverage can be considered, and in principle higher levels (i.e. the more external ones) represent more coverage regarding one specific dimension for adaptation. So, for instance, an application able to adapt regarding varied modalities (multi) can be classified as having a higher coverage level of adaptation regarding the modality dimension if compared with another application that performs adaptation within the same modality type (intra).

It is worth to notice though, that a higher coverage level of adaptation regarding one specific dimension does not immediately imply a higher level of usability or a better application for the end users. Implementing adaptation imposes many trade-offs (e.g. adapting an application may negatively affect its performance or accessibility level), and as such only by carefully planning and performing evaluation sessions, the actual benefits of adaptation for the end users can be identified.
3.2 Description

The new version of the CADS, as mentioned above, maintains the advantages of its previous version and removed some issues that could lead to misunderstandings. This section explains the characteristics of this third version, highlights its advantages and discusses its weaknesses.

As mentioned above the CADS diagram is extensible and flexible, and thus dimensions can be removed, inserted, or better detailed. Below there is a list of the dimensions included in the CADS illustrated in Figure 4. Clearly, for analyses that require more focus, a specific set among these dimensions can be selected. On the other hand, for broader analysis it is also possible to include and consider further dimensions and granularity levels.

The dimensions described below represent the basic structure for the CADS:

- **User Interface Component Granularity**: this dimension defines the levels of abstraction for the UI elements that can be subject to adaptation. Three levels are defined for these dimensions, *interactor* level, *dialog* level and *total* level. Interactors correspond to UI elements (e.g. a combobox), dialog refers to containers (i.e. a composition of UI elements), and total level refers to the complete window. It is worth to note that this dimension is only valid in principle in the context of GUIs.

- **Modality**: this dimension refers to the adaptations that change the modality type for the user interaction, when the same modality is maintained the modality level is classified as *intra-modality*, when it changes from one type to another it is *inter-modality*, and when multiple modality types are involved, the adaptation is classified as *multi-modality*.

- **State Recovery Granularity**: this dimension refers to the application of the adaptation towards the impact to the end user, i.e. if the user is obliged to quit the session and re-start a new one, the state recovery occurs at the *session* level, if the task is impacted the recovery occurs at the *task* level, and
if just the action itself is impacted, the recovery is said to be at the **action** level.

- **User Interface Deployment**: this dimension represents how much adaptation has been pre-defined at design-time vs. computed at runtime, thus respectively permitting a *static* or a *dynamic* deployment.
- **User Feedback**: this dimension refers to how the opinion of the user is taken into account, i.e. if the system is adapted, and the user can just accept or reject the adaptation after it has been performed, it can be classified as *Post*; if she is able to accept it (or reject) before it is applied, it is said to be *Pre*; evaluations refer to the possibility of the users to provide their feedback to the system, in a *numeric* (e.g. with a Likert scale) or *literally*, providing further details about their feedback.
- **Technological Space Coverage**: this dimension refers to the technologies adopted and used by the application, when the same technology is maintained it is classified as *intra*-technological space, when the technology changes from one to another, it is called *inter*, and with multiple technologies are possible, it is classified as a *multi*-technological space adaptation.
- **Existence of a Meta-UI**: consists in abstract models that formally represent and handle the adaptation process and also allow users to control, evaluate and evolve it. It comprises: *no-meta UI, meta-UI without negotiation, meta-UI with negotiation*, and *plastic meta-UI.*
- **Autonomy Levels**: it refers to the level in which adaptation is implemented, i.e. *designed* applications do not perform adaptation at all, *adaptable* applications rely on users to trigger and perform the adaptation, *adaptive* systems rely on the adaptation to be automatically performed, and *self-modifying* means evolutionary systems able to adapt their own adaptation engines.

The following dimensions were transferred to the CARF, because they are descriptive, their levels are not ordered, and their classification consists generally in being present or absent in an adaptive or adaptable application.

- **Adaptation Means**: such as re-molding and re-distribution, they can be considered as two possible adaptation techniques (or methods since they can be composed by a set of different adaptation techniques), they can be present or absent in an application, and no ordered relationship is established between them.
- **Context of Use**: this dimension refers to the branch to what of the CARF, i.e. in case of a context-aware adaptation, which is the context information taken into account to define the adaptation. Usually user, platform and environment are considered, but application domain can also be included. Although there is no ordered relationship empirically associated with them, it is possible to associate priorities to each specific context information in a given context of use, i.e. varying according to the application (or even the state, the situation, or the circumstances).

### 3.3 Application

To apply the new version of the CADS, the developer relies on the main axis to mark the coverage level for each CADS dimension. The marks are defined according to what was previously considered during the implementation of the application to be analysed and what is currently available. So, for instance if the adaptation regarding the UI component granularity occurs at the interactor level, the axis must be marked (highlighted) until this specific level. This procedure must be repeated for each dimension of the model. As a result the developer will have the applied CADS in which she can easily identify the dimensions that were better explored and the ones that could be also taken into account to extend the application in the future.

Another possibility to mark the dimensions consists in colouring (with stronger tones) the region of the circle under the level of interest, however this approach works well only if all the levels correspond to the circles, and besides comparing multiple applications would not be possible with such an approach.

In order to compare two or more applications, developers have two possible approaches: (i) parallel lines can be drawn in different colours, allowing a straightforward comparison; or (ii) an additional model can be used, comparing thus different application of the CADS in parallel. Both approaches permit multiple applications to be compared simultaneously, however for a large number of samples the second approach is recommended since it does not affect the readability of the dimensions’ labels.

Once the comparison of multiple applications rely on colour to differentiate them, it is necessary to choose then different tones or styles, avoiding thus accessibility issues that may rise in case color blind users for
example.
The Figure 5 illustrates the application of CADS to analyse a context-aware adaptation case. In this example, the UI component granularity is classified as Total, the Modality is classified as intra, the State recovery granularity as Session, the UI deployment as Static, the User feedback as a numeric evaluation, the Technological Space Coverage as intra, the Existence of a meta-UI as meta-UI without negotiation, and the Autonomy level as Adaptable.

![Figure 5. Example of the application of the CADS](image)

### 3.4 Advantages

The new version of CADS keeps the same advantages of its previous version, e.g. it is flexible because it allows developers to analyse dimensions according to their needs, i.e. they can select which dimensions will be considered in the analysis and use the diagram applying only the dimensions of interest.

The CADS illustrated in Figure 6 exemplifies the application of the model considering 6 selected dimensions. This advantage permits developers to perform more fine-grained analysis.

Since CADS is a flexible and extensible model, new dimensions can be included in the list and also applied. The main criteria to perform it is assuring that it is possible to analyse the dimensions in a ordered way, for instance by defining different granularity levels, or a scale.

One example of refinement for the Autonomy Level dimension consists in adding a Mixed-Approach level on top of Adaptive. Mixed-Approaches occur when both: end user and system are capable of taking decisions during the adaptation process.
3.5 Final Remarks

The new version of the CADS is a result of the evolution of its previous versions; its weaknesses and strengths were analysed and discussed during the meetings, presentations and also with an internal survey. The new version maintains the strong points of the preliminary versions and tries to overcome the problems regarding mainly potential misunderstandings caused by including also dimensions that are not ordered. To solve this issue, these dimensions were transferred to the CARF models, and they are described in detail in the next Section.

It is worth to notice that although the CADS establishes an empirical relation of order among the levels that compose each dimension, the concepts considered are still associated with qualitative variables, i.e. they (in principle) can not be numerically evaluated and proportionally compared. Therefore, for each application of the model it is necessary to justify the selection process and its respective application.

The main advantage of the CADS consists in the possibility to analyse adaptation in a unified view, i.e. considering simultaneously a set of dimensions and levels that are relevant in the context.
4 Context-Aware Reference Framework (CARF)

4.1 Contextualization

The CARF is defined as a reference framework that specifies the most relevant concepts to implement and perform context-aware adaptation. This reference framework has a graphical representation composed by seven branches that contains potential instances for implementing, performing and also analysing context-aware adaptation.

The graphical representation of the CARF is a mind map, in which the first level (i.e. the nodes directly connected to the root) present the abstraction of concepts and lower-level nodes detail and exemplify instances for context-aware adaptation.

In the previous release of this deliverable, D2.1.1, the CARF was explained and we focused on the how branch detailing an extensive list of adaptation techniques by resource type (i.e. content, presentation and navigation). These techniques were presented by means of templates that described further information about them. Since then, many other techniques were identified and detailed, for instance the technique to adapt (change) the font type for users with dyslexia, as Figure 7 exemplifies.

<table>
<thead>
<tr>
<th>Name</th>
<th>Dislexie</th>
</tr>
</thead>
<tbody>
<tr>
<td>References</td>
<td><a href="http://www.studiostudio.nl/">http://www.studiostudio.nl/</a></td>
</tr>
<tr>
<td>Description</td>
<td>Given a text content its font type will be modified for users with dyslexia</td>
</tr>
<tr>
<td>Rationale</td>
<td>Given a text content, its presentation will be modified</td>
</tr>
<tr>
<td>Example</td>
<td>The user is reading an article of the news and it has dyslexia, the font type is modified in order to provide higher accessibility levels</td>
</tr>
<tr>
<td>Context</td>
<td>User’s impairments (dyslexia)</td>
</tr>
<tr>
<td>Advantages</td>
<td>The reading will be possible, improving the usability and accessibility levels</td>
</tr>
<tr>
<td>Disadvantages</td>
<td>It may decrease the performance (processing before rendering the text content)</td>
</tr>
</tbody>
</table>

![Figure 7. Template describing the adaptation technique of changing the font type for users with dyslexia](image)

The templates describing adaptation techniques were used as a base to formalize adaptation rules and also to specify the algorithms presented by the D4.2.1 (Library of Algorithms). The abstract concepts defined by the CARF also supported the creation of the meta model for context-aware adaptation, presented in the D3.1.1.

Since during the last review meeting the coverage of adaptation techniques was considered good, and sufficiently broad, in this section we will specially focus on the other CARF branches, so instead of emphasizing details about all adaptation techniques, we detail the other concepts and describe the application of the CARF to analyse adaptive applications, as the demonstrator.
4.2 Description

The root of the CARF is composed by seven branches, as Figure 8 illustrates, in clockwise sense they are: why, what, who, when, where, to what, and how. These branches are abstract concepts defined as follows:

- **Why**: defines the main goals for the adaptation process, they are mainly expressed in terms of software qualities. For example: adaptation can be performed aiming to improve the usability levels;
- **What**: represents the type of resources that is adapted, in general they can be defined as navigational flow, presentation or content. For example: re-size all the images;
- **Who**: refers to the actor who triggers and is in charge of each phase of the adaptation process, for instance the end user, the system, or a third party. For example: in mixed approaches both end users and system collaborate with the adaptation;
- **When**: represents the state in which the adaptation process is performed, i.e. it can occur at design time, run time, compilation time. For example: adaptations performed at run time;
- **Where**: this branch refers to the virtual location in which the adaptation take place, i.e. according to the architectural approach adopted it can be at the client, at the proxy, or at the server. For example: adaptation performed at the server side;
- **To what**: lists potential context information that justifies and defines the adaptation process, i.e. usually the application resources are subject to adaptation according to the user, the platform, or the environment. For example: adapting according to colour-blind users;
- **How**: defines how the adaptation process is performed, by listing possible methods, techniques and strategies for adaptation. For example: change the font size.

These seven branches compose the core of the CARF, and by adding new instances they can be refined, however they cannot be extended with additional branches (since these concepts are fixed and they were previously defined as the most relevant ones, and they are sufficient to comprise the complete phases of an adaptation process).

The main goal of the CARF consists in defining the most relevant concepts for adaptation and to extensively list and present the possibilities regarding its implementation and execution. The CARF can be used before the implementation phase of an application, as an extensive catalogue to guide developers in taking design decisions, or after the implementation phase of an application, to analyse or evaluate the concepts that were taken into account (and consequently identify possibilities for future extensions).

In order to include an extensive list of instances in the CARF, separate models were generated according to the type of resource subject to adaptation, separating the models by resource types helps to edit and update them more quickly. The final goal though is to join all the concepts and generate one single model containing all of them.

Considering that the updated version of the CARF includes hundreds of concepts, we opt to represent them in this deliverable in an alternative format, i.e. due to readability issues, for the constrained dimensions of the deliverable, we will list all the concepts in the following sessions, and keep their CARF representation for the digital version of the diagram, which is interactive, and permits to zoom in, out, as well as collapse and expand the branches.
4.3 How

4.3.1 Adaptation Techniques and Methods

Adaptation techniques consist of actions that are performed in application resources to modify them or their specific properties. Methods are a composition of adaptation techniques that are compatible and thus can be jointly applied to modify one or more resources. The Table 1 lists all techniques and methods gathered by the CARF⁴. They are organized by resource type.

4.3.2 Adaptation Strategies

Strategies for adaption consist in approach types used to implement the adaptation process.

From a technical perspective they can be classified as:

- **Graceful Degradation**: when the application is able to adapt itself according to the constraints imposed by the context of use, and thus reduce its functionalities or resources properties. Examples include: disabling scripts according to the browser type or version, and replacing videos by equivalent text descriptions depending on the connection speed available.

- **Progressive Enhancement**: consists in providing further options for the end user, in terms of functionality, contents or resources properties, according to her context of use. For instance, users with large screens are able to access more contents at the same moment, this characteristic of the context can be then appropriately explored by the adaptation (i.e. by optimizing the use of all end user resources and consequently providing them a better experience).

From the end user perspective, the results of the adaptation can be presented using different approaches. Since usually users can feel confused with disruptive changes between the original UI and the adapted one, animation for instance can be applied aiming to help them to follow and to understand better such changes.

Animation approaches include: brighten, collapsing, cross fading, dim, expanding, fading in, fading out, morphing, progressive rendering, self healing, sliding, spotlighting, and plugging in and plugging out components.

---

⁴ As mentioned before, we opt to use an alternative representation instead of CARF aiming to assure the readability of the concepts in the printed version of this document.
<table>
<thead>
<tr>
<th>Navigation</th>
<th>Presentation</th>
<th>Content</th>
<th>Audio</th>
<th>Image</th>
<th>Text</th>
<th>UI Element</th>
<th>Video</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessible Navigation</td>
<td>Attach</td>
<td>Change Direction</td>
<td>Change Bit Rate</td>
<td>Adjust Shape</td>
<td>Additional Explanation</td>
<td>Change Size</td>
<td>Change Frame Resolution</td>
</tr>
<tr>
<td>Annotate Links</td>
<td>Augment</td>
<td>Conditional Text</td>
<td>Change Sample Rate</td>
<td>Collapse to zoom</td>
<td>Align</td>
<td>Replace</td>
<td>Change Frame Spatial Quality</td>
</tr>
<tr>
<td>Organize Alphanumerically</td>
<td>Collapse to Zoom</td>
<td>Frame Based</td>
<td>Convert Channel</td>
<td>Change Brightness</td>
<td>Alter Fragments</td>
<td>Adapt Form</td>
<td>Skip Frame</td>
</tr>
<tr>
<td>Navigate Alternately</td>
<td>Detach</td>
<td>Filter</td>
<td>Convert Format</td>
<td>Change Color Balance</td>
<td>Auto-correct</td>
<td>Adjust Form</td>
<td>Semantic Summary</td>
</tr>
<tr>
<td>Organize Chronologically</td>
<td>Distribute</td>
<td>Segment by Index</td>
<td>Translate Language</td>
<td>Change Color Map</td>
<td>Change Background</td>
<td>Expand TextBox</td>
<td>Remove Shot</td>
</tr>
<tr>
<td>Concern-Sensitive Navigation</td>
<td>Fish eye</td>
<td>Personalize</td>
<td>Translate Modality</td>
<td>Translate Color</td>
<td>Change Color</td>
<td>Split Table</td>
<td>Transcode Format</td>
</tr>
<tr>
<td>Categorize</td>
<td>Full-screen</td>
<td>Re-size</td>
<td>Change Volume</td>
<td>Change Contrast</td>
<td>Explain Comparatively</td>
<td>Transform Table</td>
<td>Select</td>
</tr>
<tr>
<td>Guide Directly</td>
<td>Large Screen</td>
<td>Re-order</td>
<td>Change Speed</td>
<td>Change Quantization</td>
<td>Compare</td>
<td>Split Interface</td>
<td>Reduce</td>
</tr>
<tr>
<td>Disable Link</td>
<td>Migrate</td>
<td>Stretch</td>
<td>Truncate</td>
<td>Change Transparency</td>
<td>Change Contrast</td>
<td>Moving Interface</td>
<td>Replace</td>
</tr>
<tr>
<td>Elise First Sentence</td>
<td>Mirror</td>
<td>Suggest (Recommend)</td>
<td>Simplify</td>
<td>Digital Compose</td>
<td>Describe</td>
<td>Visual Popout Interface</td>
<td>Synthesize</td>
</tr>
<tr>
<td>Generate Links</td>
<td>No-script</td>
<td>Translate to Vocal</td>
<td>Summarize</td>
<td>Matte</td>
<td>Dim Fragments</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hide Links</td>
<td>Optimize</td>
<td></td>
<td></td>
<td>Optical</td>
<td>Dislexie</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organize Hierarchically (within a page)</td>
<td>Overview + Detail</td>
<td></td>
<td></td>
<td>Daltonize</td>
<td>Explain Variants</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hierarchical Structure (within an application)</td>
<td>Print</td>
<td></td>
<td></td>
<td>Demosaicing</td>
<td>Change Font Family</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Segment by Index</td>
<td>Re-distribute</td>
<td>Differentiate</td>
<td>Change Font Style</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------</td>
<td>---------------</td>
<td>-------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linear Structure</td>
<td>Re-mold</td>
<td>Enlarge</td>
<td>Highlight</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adapt Map</td>
<td>Re-size</td>
<td>Elise Image</td>
<td>Orient</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Reduce)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mesh Structure</td>
<td>Replace</td>
<td>Bayer Filter</td>
<td>Outline</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Outline</td>
<td>Single Column</td>
<td>Hash</td>
<td>Geometrically</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paginate</td>
<td>Expand Textbox</td>
<td>High Dynamic</td>
<td>Pre-requisite</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Range Imaging</td>
<td>(Explain)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organize by</td>
<td>Transcode</td>
<td>Register Image</td>
<td>Change Serum</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Popularity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remove Link</td>
<td>Decompose</td>
<td>Morph Image</td>
<td>Similarity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search</td>
<td></td>
<td>Recognize Image</td>
<td>Simplify</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simplify Input</td>
<td></td>
<td>Segment Image</td>
<td>Re-Size</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Controls</td>
<td></td>
<td>Interpolate</td>
<td>Space</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sort Links</td>
<td></td>
<td>Reduce</td>
<td>Stretch Text</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rotate</td>
<td>Summarize</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Scale</td>
<td>Sort</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SmartView</td>
<td>Translate</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WebThumb</td>
<td>Truncate</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Binary Transcode</td>
<td>Zoom</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.4 To What

This branch of CARF refers to the context information that is taken into account to perform adaptation. Although context information involves all relevant information that can be useful to implement an application, it is usually classified concerning three main classes, namely the user, the platform, and the environment.

4.4.1 User

Although individual users differ on various dimensions, for most systems they still need to adjust their behaviour and problem solving strategies to efficiently interact. The systems are, in general, designed for the average user, but no for all users. On the other hand, an ideal system should adapt itself according to the current user, thus compensating their weaknesses, providing appropriate help, and decreasing her mental and physical workload. Furthermore, an adaptive system should be able to characterize and distinguish individuals. User models can support this task, by defining their knowledge, capabilities, preferences, cognitive strengths and limitations [Norcio and Stanley, 1989].

Context information regarding the user characteristics involves an extensive list of concepts. In general they can be classified as:

- Identification
- Interests and Preferences
- Educational Level
- Disabilities
- Profession
- Living
- Psychosocial State
- Socio-economic Class
- User State
- Interaction History / Usage Habits
- Health State
- Abilities
- Skills
- Demographic

For each of the concepts mentioned above, many sub-concepts can be associated. For instance the disabilities include: motor, cognitive, behavioural, visual, hearing, and speech.

Since the CARF scope does not include retrieving all possible context information regarding the user, we refer to existing works already reported in the literature to recover further context information that can be relevant within the context of Serenoa.

For the moment, just the most relevant aspects for context-aware adaptation were considered in the CARF model regarding the context of the user. Related works containing additional information can be accessed in [Brossard et al., 2010], [UMO, 2003], [Veritas].

4.4.2 Platform

The Platform is modeled in terms of resources, which determine the way information is computed, transmitted, rendered, and manipulated by users. Examples of resources include memory size, network bandwidth, and input and output interaction devices. Resources motivate the choice for a set of input and output modalities and, for each modality, the amount of information made available [Calvary et al., 2002].

The platform branch changes dynamically over time, and as such the CARF coverage considers also just the most relevant concepts regarding context-aware adaptation. The items below illustrate these concepts:

- Interaction Modalities
- Operating Systems
- Input Devices
- Output Devices
The concepts mentioned above can also be refined with related sub-concepts. The device type for instance include: Desktop PCs, e-readers, netbooks, iDTVs, laptops, mobile phones, notebooks, smartphones, tablet PCs, and PDAs.

Many works have been dedicated to define a concrete modelling about context information regarding the platform characteristics. Examples include: [W3C's Delivery Context Ontology], [OpenDDR].

### 4.4.3 Environment

Environment describes the situation and the environment in which the interaction takes place [Brossard, 2010]. According to Zimmermann (2007) environmental differences result from the mobility of computing devices, applications and people, which leads to highly dynamic computing environments. Unlike desktop applications, which rely on a carefully configured and largely static set of resources, ubiquitous computing applications are subject to changes in available resources such as network connectivity and input and output devices. Moreover, they are frequently required to cooperate spontaneously and opportunistically with previously unknown software services in order to accomplish tasks on behalf of users. Thus, the environment surrounding an application and its user is a major source to justify adaptation operations.

According to Coutaz and Rey (2002), the environment denotes the set of objects, persons and events that are peripheral to the current activity but that may have an impact on the system and/or users behavior. As such, the environment may encompass the entire world. In practice, the boundary can be defined by domain analysts who elicit the relevant entities for each case. Specific examples are: user's location, ambient sound, lighting or weather conditions, present networks, nearby objects, user's social networks, level of stress.

Although commonly neglected, the context information concerning the characteristics of the environment in which the user is located is equally important. Relevant concepts include:

- Location
- External Events
- Presence and Arrangements
- Time

Many sub-concepts for each of the concepts mentioned above can also be considered. Concerning the presence and arrangements branch, the following sub-concepts are involved: entities, artefacts, natural objects, and people.

In order to retrieve more knowledge about this context information, the following works can be consulted: [Zimmermann, 2007], [Coutaz and Rey, 2002].

### 4.5 Where

Different architectural approaches can be adopted to implement context-aware adaptation, in the case of a client-server architecture a separate software client is developed for each device, which then communicates and interprets information from a dedicated, remote, server software application, in order to show a GUI [Mitrovic, 2007]. In this context, the location in which the adaptation is executed can be defined as:

- Client
- Proxy
- Server

Examples of the proxy architectural approach can be found in [Bickmore, 1999] and in [Buyukkokten et al., 2000].
4.6 Why

The branch Why of the CARF refers to the software quality that is aimed for the adaptation process. This concept is also aligned with the non-functional requirements and with the evaluation criteria of Serenoa. In a high level overview, the qualities considered include:

- Functionality
- Maintainability
- Portability
- Reliability
- Usability
- Consistency
- Extensibility
- Security

Each of the concepts mentioned above can be refined by related sub-concepts. For instance regarding the usability as a quality, the following sub-concepts (as defined in the D2.4.1) are closely associated: accessibility, attractiveness, controllability, comprehensibility, error tolerance, feedback, flexibility, preview, learnability, memorability, satisfaction, control, effectiveness, efficiency, and collaboration.

The why branch guides the adaptation definition, for instance the adaptation process can be oriented to improve the accessibility level of an application. It can be used before implementation (e.g. to define the requirements), and after the implementation (e.g. in the evaluation phase).

4.7 What

Many different resource types can be subjected to adaptation. Generally three main groups of resources are defined [Brusilovsky, 2001]:

- Content
  - Audio
  - Image
  - Text
  - UI Elements
  - Video
- Navigation
- Presentation

These concepts can be also refined with additional sub-concepts. The UI Elements for instance include: textbox, combobox, radio buttons, forms, buttons, labels, etc. Furthermore the sub-concepts have properties that can also be targeted by the adaptation process.

4.8 Who

The branch who of the CARF defines the possible agents that trigger an adaptation process, and who are responsible for its subsequent phases too. The main actors include:

- Developer
- End user
- System
- Third-party

It is worth to mention that given the multiple phases that compose an adaptation process it is likely that multiple agents are involved collaborating in the decisions of adaptation. Besides, the actors involved permit the application to be classified as: Adaptable (when the user is responsible), Adaptive (when the system is responsible) or Mixed-Approach (when multiple entities are involved).

4.9 When

The adaptation can be performed in different phases of an adaptation process. These phases include:
4.10 Application

Stakeholders can use the CARF before implementing an application, as an extensive catalogue of possibilities to perform context-aware adaptation. Besides, during the development life-cycle it can also be used to verify alternative options for implementation. And after the application was implemented, in order to analyse the branches that were considered, in which extent, and also to evaluate its coverage level and to compare applications. By detecting the possibilities that were not initially considered, stakeholders can identify further opportunities for extending and updating their applications.

Although the CARF achieved a stable version concerning its branches, it is possible to extend and refine the model with more instances, updating it according to the progress and evolution of new technologies.

4.10.1 Applied CARF

The current version of the demonstrator application was taken as an example to apply the CARF. Thus, this current version the CARF, illustrated by Figure 10, is composed as follows: how consists in changing image colours, changing modality type, changing font type; to what consists of platform, user and environment; where is at the server; why is aiming to improve the usability levels; what consists of presentation, contents (as images and texts), and navigation; who consists of developers, end user, and system; and when is at design time.

![Diagram](image)

Figure 10. CARF applied.

4.11 Final Remarks

The Context-Aware Reference Framework (CARF) provides an extensive list of concepts that can be used for stakeholders to propose, implement and analyse adaptive and adaptable applications. The coverage of the concepts is considered extensive, and the CARF supports future refinements and extensions to be updated according to the evolutions in its application domain.

The application of the CARF supports stakeholders during the complete development life-cycle of applications.

Once the CARF defines the concepts in different abstraction levels, it provides also a base to define meta models for context-aware adaptation; as exemplified in the D3.1.1.
5 Conclusion

Together, the CARF, CADS and CARFO constitute a computational framework for the development of context-aware adaptation in technological applications.

5.1 Final Remarks

The Theoretical models of Serenoa tackle an important gap in the scientific literature, since so far the stakeholders wishing to implement context-aware adaptation had no theoretical methods to support them during the complete development life-cycle of such applications.

Both CADS and CARF are complementary methods that support stakeholders in taking design and implementation decisions. Besides they cover the different phases of the development life-cycle, as design, implementation, tests and evaluation.

The models are extensible, which permits them to follow the dynamic updates that are commonly found in the technological domain.

All the concepts were deeply discussed among all partners until a stable version of the models was achieved. Besides, even during their development the CADS model was adopted and applied.

5.2 Future Work

As a future work we envision the implementation of the CADS and the CARF as online tools, in which the stakeholders can submit their specifications and have the graphical representations generated in order to perform the analysis of their applications.

Besides, as a future work we plan to express information regarding the relevant concepts of the CARF and the CADS in the CARFO Ontology, such as the information in Table 1. In this way, a RUIGE sub-module might have a manifest associated, which would make explicit which adaptation techniques and methods per resource type it supports, to what users it is devoted (target audience), target platforms (Interaction Modalities, Operating Systems, Input Devices, Output Devices, Browser, Device Type, Device Properties – note that target platforms can be solved by means of a query to the Context of Use Module and, potentially, to an underlying DDR technology), environment concepts handled (Location, External Events, Presence and Arrangements, Time, …), where each type of adaptation takes place (client-side, server-side, intermediate proxy), what resources are subject to adaptation, who triggers the different adaptation processes (Developer, End user, System, Third-party) and when the distinct adaptation tasks take place (design time, compilation time, runtime). In this case, the CARF/CADS module of the CARFO should be ready for the R2 of the CARFO deliverable. The manifest for each RUIGE sub-module would allow querying for the capabilities of each module and a module (orchestrator) might allow users/programmers to know which RUIGE sub-module(s) serve their needs.
References


[Veritas] http://veritas-project.eu/


Acknowledgements

- TELEFÓNICA INVESTIGACIÓN Y DESARROLLO, http://www.tid.es
- UNIVERSITE CATHOLIQUE DE LOUVAIN, http://www.uclouvain.be
- ISTI, http://giovce.isti.cnr.it
- GEIE ERCIM, http://www.ercim.eu
- W4, http://w4global.com
- FUNDACION CTIC http://www.fundacionctic.org
Glossary

http://serenoa.morfeo-project.org/glossary-of-terms